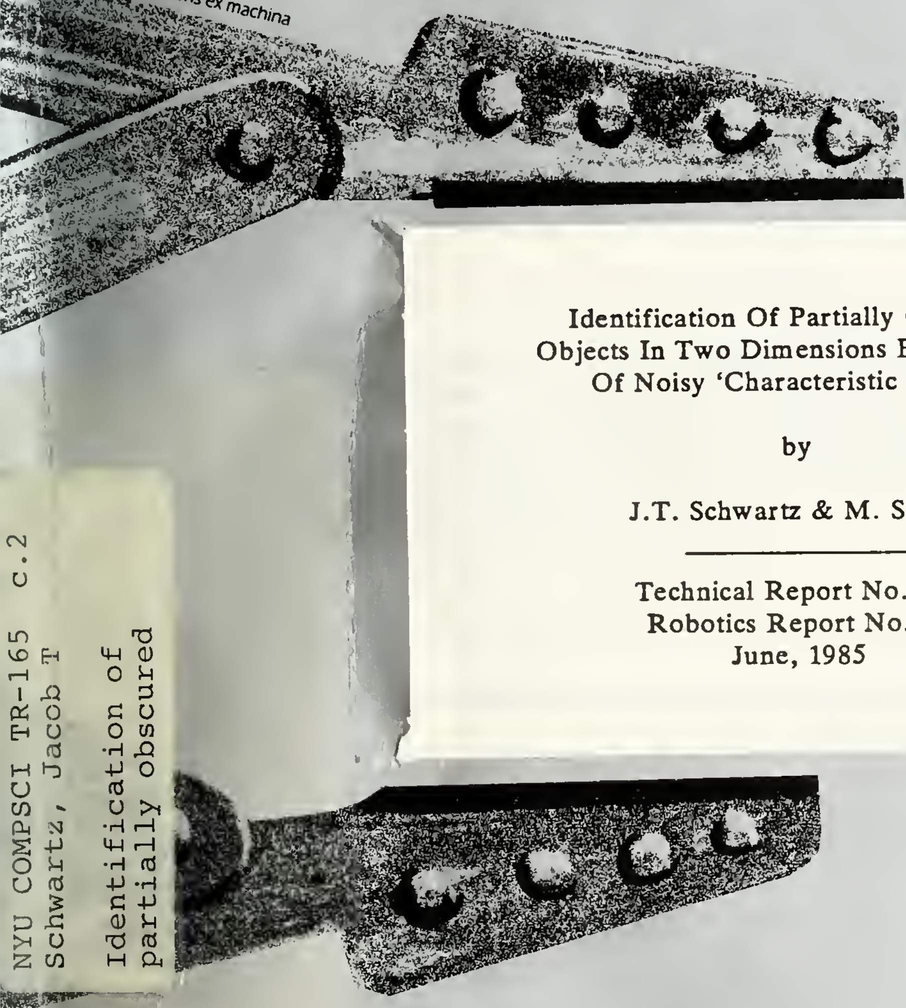
Robotics Research Technical Report

Generatorium omnis laboris ex machina



Identification Of Partially Obscured Objects In Two Dimensions By Matching Of Noisy 'Characteristic Curves'

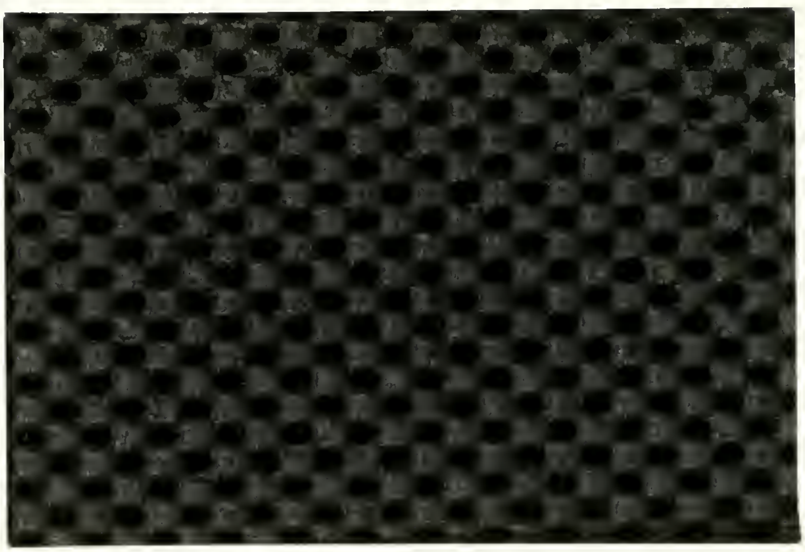
by

J.T. Schwartz & M. Sharir

Technical Report No. 165
Robotics Report No. 46
June, 1985

New York University
Courant Institute of Mathematical Sciences

Computer Science Division
251 Mercer Street New York, N.Y. 10012



Identification Of Partially Obscured Objects In Two Dimensions By Matching Of Noisy 'Characteristic Curves'

by

J.T. Schwartz & M. Sharir

Technical Report No. 165
Robotics Report No. 46
June, 1985

New York University
Dept. of Computer Science
Courant Institute of Mathematical Sciences
251 Mercer Street
New York, New York 10012

Work on this paper has been supported by Office of Naval Research Grant N00014-82-K-0381, National Science Foundation Grant No. NSF-DCR-83-20085, and by grants from the Digital Equipment Corporation, the IBM Corporation, and the US-Israeli Binational Science Foundation.

1.2.

Identification of Partially Obscured Objects in Two and Three Dimensions by Matching of Noisy 'Characteristic Curves'

Jacob T. Schwartz and Micha Sharir

Computer Science Department

Courant Institute of Mathematical Sciences

New York University

ABSTRACT

The problem of matching partially obscured noise-corrupted images of composite scenes in two and three dimensions is analyzed. We describe efficient methods for smoothing the noisy data and for matching portions of the observed object boundaries (or of characteristic curves lying on bounding surfaces of 3-D objects) to pre-stored models. Initial experiments showing the efficacy of this procedure are reported.

1. Introduction

The 'object identification' or 'model-based vision' problem can be defined as follows: We are given an image of an object O, which is assumed to be a member of a collection of possible objects $O_1, \ldots, O_n$ known *a priori* (e.g. $O_1, \ldots, O_n$ may be a collection of parts involved in an assembly/disassembly process, in an industrial environment in which only objects drawn from the set $O_1, \ldots, O_n$ can legitimately appear.) However, the object O is observed from an unknown angle, and moreover only part of O is observed, the rest of its surface being 'obscured' in some manner (usually by other objects belonging to the family $O_1, \ldots, O_n$ which happen to overlap O; see below). Our problem is to identify the object O as one of the elements in the list of 'model' objects $O_1, \ldots, O_n$ given *a priori* and to

determine the position and orientation at which O is being viewed.

This is the *object identification problem* for a single object. We may also wish to consider the more difficult identification problem for compound scenes ('bin image' problem), in which the image to be analyzed is allowed to contain several objects $O, O', O'' \cdots$ all drawn from the specified collection $O_1, \ldots, O_n$ of model objects, but viewed from an angle in which the images of $O, O', O'' \cdots$ can overlap and can obscure parts of each other.

This problem is fundamental to computer vision and has been discussed repeatedly in the literature; see [BB82] for a review of standard approaches. The approach to be described differs from the feature-based techniques commonly used (see, e.g. [BC82] and [BHH84]) in that it matches observed objects directly to models but in a more robust and efficient manner than other related schemes that have been reported (see [ABBF84], [KK85], [TMV85]) in that it makes only very limited use of tangent direction estimates and uses high-speed algorithms to calculate the rotation and translation which best match an observed object to a prestored model. The approach used bears some relation to that described in [GLP84], [GLP85]. but is independent of any assumption that the objects to be identified should be polygonal or polyhedral.

The present paper will consider the identification problem largely in two dimensions, but with some initial attention to the somewhat more difficult three-dimensional case. In both two and three dimensions we will make the simplifying assumption that images which represent the observed objects $O, O', O'' \cdots$ in their true geometric size are available. In the 2-D case, this would be the case for objects cut out of cardboard or sheet metal, observed by a camera a known fixed distance above them. In the 3-D case, the 'fixed-scale' images assumed can be produced with the aid of a 'depth' sensor which produces an 'image' showing the true 3-D position of each point in every observed body surface. Camera-based depth sensors of this kind are now becoming common; see [S83, BS84] for an account of one such sensor.

Through Section 6 below discusses surface-based methods for object identification; the principal approach to the object identification problem that will be outlined in the present paper is based upon a notion of 'rigidly embedded curve'. This is any curve C defined by the geometry of an object O and fixed in O, so that when O undergoes any rotation and translation (i.e. Euclidean motion) the curve C moves with it. For 2-D objects, these curves can simply be the object boundaries; for 3-D objects, other somewhat more subtly defined curves, for example curves of specified constant principal curvature on the observed surface, or lines of maximum curvature ('ridges') can be used instead. (A somewhat more detailed discussion of various potentially useful characteristic curves for 3-D object surfaces is found in Section 4 below.)

The main property that we require these curves to possess is that they should be relatively immune to noise. That is, if we take a model object O with a characteristic curve C on it, and then perturb O by some random noise of maximum size ϵ to obtain a 'noisy' image O' of it, then the corresponding curve C' on O' should not deviate from C by more than some function of ϵ which tends to zero with ϵ. As we shall see below, to enforce such behavior, some preliminary 'smoothing' will be needed before C' is calculated. More details concerning a smoothing technique which seems appropriate are given in Section 2 below.

Once we have defined the curves to be used, our proposed identification method proceeds as follows:

(a) The characteristic curves C on the observed object and on all the model objects are parametrized by arc length s, giving (vector valued) functions $c(s)$ (for the observed object) and $c_i(s)$ (for the various model objects).

(b) We find the offset s_0, the index i, and the Euclidean motion E which cause the function $Ec(s)$ to match the model curve $c_i(s+s_0)$ most closely (the offset s_0 is needed in cases in which, because of obscuration, only a portion of the curve C can be obtained from the viewed image). The orientation of O' (relative to the model object O which it matches) is

then defined by the transformation E. We shall show that determination of E and s_0 can be accomplished rapidly if we measure the distance from $c(s)$ to $c_i(s+s_0)$ in a least-squares sense. Indeed, this mode of approximation will be seen to allow E to be calculated explicitly and s_0 to be determined by a fast Fourier-transform based method, and with the happy consequence that the total cost of matching is only $O(nm \log m)$, n being the total number of model objects (or more precisely the total number of characteristic curves), and m the number of (evenly-spaced) points of interpolation introduced along each curve to approximate it. (A hashing technique for removing the factor n from this last cost formula is described in the companion paper [KSSS85].)

(c) As already noted, observation of any characteristic curve C of the viewed image O' of some model object O will always produce a variant C^* of C that is perturbed by addition of a certain amount of noise. Since this noise can introduce irregularities in C that distort its arc length significantly, matching must be preceded by a smoothing operation which takes C^* and brings it back closer to the original curve C.

We now proceed to explain how all these various steps can be accomplished.

2. Smoothing Noise-Distorted Curves by Finding Shortest Local Approximations

Existing methods for smoothing noise-corrupted curves ordinarily use some band-pass convolution operation. However, we will argue that in preparing for curve-matching a rather different approach, which smooths such curves by finding shortest paths lying in the neighborhood of an observed noisy curve, can be preferable. To this end, we begin by giving the following simple

Definition 2.1: Let c be a parametrized curve in Euclidean n-space E^n, and let L be its length. Then

(i) The ϵ-neighborhood $U_\epsilon(c)$ of c is the set of all points in E^n whose distance from some point of c is at most ϵ.

(ii) The curve c is (ϵ,δ)-stable if every curve in $U_\epsilon(c)$ which connects the endpoints of c has length at least $L - \delta$.

It is clear that every simple (i.e. non self-intersecting) smooth curve is $(0,0)$-stable, and thus will remain (ϵ,δ)-stable for every positive δ and sufficiently small ϵ.

Suppose next that the curve c^* is obtained from c by addition of some perturbing (noise) function of maximum magnitude of at most ϵ, so that c^* lies in $U_\epsilon(c)$ and connects two points a, b that lie within ϵ of the corresponding endpoints of c. We shall suppose that all the curves with which we deal, here and below, are parametrized in terms of their arc length s, and that c^* is differentiable. To smooth c^*, what we want to do is to derive another parametrized curve $c'(s)$ from it such that $|c(s)-c'(s)|$ is uniformly small over the whole length of c; once this has been done, it becomes possible to use the matching procedure sketched in (b) above to match the curve c' to the model curve c.

For this we define c' to be the shortest curve in $U_\epsilon(c^*)$ connecting the endpoints of c^*. To show that c' has the required property, we can reason as follows. Plainly c' lies within $U_{2\epsilon}(c)$ and c lies within $U_\epsilon(c^*)$. Suppose that c is $(2\epsilon,\delta)$-stable, and let L' (resp. L) be the length of c' (resp. c). Then by definition $L' \geq L - \delta$. Moreover, the endpoints a, b of c^* can be connected by a curve which goes from a to its nearest point on c, then along c to a point nearest to b, and finally from there to b. Since c lies within $U_\epsilon(c^*)$ and c' is the shortest curve in $U_\epsilon(c^*)$ with the same endpoints, we must have $L + 2\epsilon \geq L'$. Thus $|L - L'| \leq \max(\delta, 2\epsilon)$.

Take a point $c'(s)$ along c'. This plainly lies within a distance 2ϵ of some point $c(s_1)$ of c. Note that we can connect the initial point of c to its final point by going along c to $c(s_1)$, then along a straight segment to $c'(s)$, then along c' to a point nearest to the final point b of c, and finally along a straight segment to b. This whole curve plainly lies within $U_{2\epsilon}(c)$ and its length is at most

$$s_1 + L' - s + 4\epsilon \leq L + s_1 - s + 6\epsilon \ .$$

Thus we must have $L - \delta \leq L + 6\epsilon + (s_1 - s)$, i.e. $s - s_1 \leq 6\epsilon + \delta$. Similarly, we can

connect the final point of c to its initial point by going back along c to $c(s_1)$, then along a straight edge to $c'(s)$, then to the point on c' nearest to the initial point a of c, and finally along a straight edge to a. Arguing as before we find that $L - \delta \le L + 4\epsilon + (s - s_1)$, i.e. $s_1 - s \le 4\epsilon + \delta$. This shows that $|s - s_1| \le 6\epsilon + \delta$, from which the following lemma follows at once.

Lemma 2.2: If the curve $c(s)$ is $(2\epsilon,\delta)$-stable, c^* is derived from c by adding a noise signal of maximum modulus at most ϵ, c' is the shortest curve within $U_\epsilon(c^*)$ connecting the endpoints of c^*, and all curves are parametrized by their arc-length, then $|c'(s) - c(s)| < 8\epsilon + \delta$ for all s in the common domain of c and c'. Moreover the lengths of c and of c' differ by at most max $(\delta,2\epsilon)$.

This lemma shows that, if c is stable in the sense of Definition 2.1, with sufficiently small ϵ and δ, the curve c' must match c well in the sense that we require.

To be sure that c' can be obtained efficiently from the observed curve c^* we must now show how to calculate shortest paths lying in the neighborhood of a given noisy polygonal curve c^* and connecting the endpoints of c^*.

An algorithm for finding shortest paths near a given polygonal curve in two and more dimensions.

We first consider the relatively simple case of plane curves, and begin by considering the restricted case in which c is a polygonal (plane) curve which is the graph of a piecewise linear function $y = c(x)$, $x \in [a,b]$. Let $c_1(x)$, $c_2(x)$ be two additional piecewise linear functions satisfying $c_1(x) < c(x) < c_2(x)$ for each $a \le x \le b$, and let U denote the closed polygonal strip between c_1 and c_2, i.e.

$$U = \{(x,y) : a \le x \le b,\ c_1(x) \le y \le c_2(x)\} .$$

(Note that in smoothing a polygonal graph c^* as in the preceding section, we can take $c_1(x) = c^*(x) - \epsilon$, and $c_2(x) = c^*(x) + \epsilon$, $x \in [a,b]$.) Let $A = (a,c(a))$ and let $B = (b,c(b))$ be the initial and final points of c. Our goal is to find the shortest (necessarily

polygonal) path p within U connecting A to B. The technique presented below achieves this in time linear in the number of corners of U. It is similar to a more general but more complex linear-time technique for finding shortest paths with simple polygons due to Lee and Preparata [LP84].

For each point $z \in U$ let $\pi(z)$ denote the shortest path from A to z within U. It is clear that in the restricted situation considered each of the paths $\pi(z)$ must be a polygonal path which is the graph of a single-valued function of x, whose corners must lie in the set N consisting of the points A and B, together with all corners of the graphs of c_1, c_2. For each point $z \in U$ let $\nu(z) \in N$ denote the initial endpoint of the last straight segment in $\pi(z)$. To obtain p we will compute the map ν for all corners in N by using the following left-to-right sweeping technique. For each $t \in [a,b]$ let $l(t)$ denote the vertical line $x = t$, and let $Q(t)$ denote a list of disjoint open intervals $I_1, \ldots, I_j$ along $l(t)$ having the following properties:

(i) The union of the closures of the intervals in $Q(t)$ contains $l(t) \cap U$, and in the list Q(t) the intervals $I_1,\ldots,I_j$ occur in ascending vertical order.

(ii) For each interval I_k there exists a unique point $w \in N$ such that for all $z \in I_k$ we have $\nu(z) = w$. Let us agree to designate this property by saying that I_k is contained in the *zone of influence* of the corner w.

Let I be one of the intervals in $Q(t)$, and assume that it is contained in the zone of influence of some $w \in N$. Let L_1 and L_2 be the two lines connecting w to the two endpoints of I. Then, if t' is slightly larger than t and no corner in N has an abscissa lying in $[t,t']$, it is easily checked that the intersection of $l(t')$ with the wedge bounded between L_1 and L_2 will yield an interval I' in $Q(t')$ which is contained in the zone of influence of w, and furthermore all intervals in $Q(t')$ are obtained in a similar manner from the intervals in $Q(t)$. In other words, if we let $l(t)$ move rightward, then, as long as it does not pass through a corner in N, the combinatorial structure of $Q(t)$ remains unchanged, and the endpoints of each interval I in $Q(t)$ simply move along straight rays which emerge from the point $w \in N$ in whose zone of influence I lies.

Next suppose that for some t_0, the line $l(t_0)$ does pass through a point $z \in N$, and without loss of generality suppose that z belongs to the graph of c_1. Suppose that just before reaching t_0, the list $Q(t)$ consisted of the intervals $I_1, \ldots, I_m$. Let $w = \nu(z)$, and let I_j be that interval which is contained in the zone of influence of w. Then if $l(t)$ is moved slightly to the right of t_0, the list $Q(t)$ will have one of the following two forms:

(i) Let e be the straight segment in the graph of c_1 whose left endpoint is z. If e lies below the line L passing through w and z, then the list Q will consist of the intervals $I_0, I_j^*, I_{j+1}, \ldots, I_m$, where I_0 is the intersection of $l(t)$ with the triangular wedge (having z as apex) bounded by e and L, and I_j^* is the intersection of $l(t)$ with the wedge bounded by L and the line connecting w to the upper endpoint of I_j. In other words, a lower portion of I_j is truncated at z, a new interval I_0 is added below it, and the intervals $I_1, \ldots, I_{j-1}$ below I_j are all removed from the list Q. Note that I_0 belongs to the (new) zone of influence of z, whereas I_j^* is in the zone of influence of w.

(ii) If the edge e lies above L, Q will consist only of $I_j^*, \ldots, I_m$, where I_j^* is defined as above. (Note that in this case I_j^* is not entirely contained in U; however by allowing I_j^* to intersect the exterior of U in such cases, we simplify the handling of the list Q without losing accuracy of the shortest path tracing which is what we require. A similar observation applies to cases in which no vertex is encountered, but in which the upper and/or lower boundaries of U progressively invade and cut off more and more intervals of the list Q.)

Symmetric rules govern the case where z lies on the upper boundary c_2 of U.

These observations allow us to maintain the list $Q(t)$ in the following manner: Intervals of Q can be associated with the points in N in whose zone of influence they lie and their endpoints represented by the coefficients of the rays on which these endpoints lie. The resulting combinatorial description changes only when $l(t)$ passes through one of the corners of N, which we can suppose to have been presorted according to their x-coordinates in left to right order. We process each $z \in N$ in turn. If z lies on c_1, we search Q in linear order from

its lowest interval upwards, until the interval I containing z is found. Let w be the point in N in whose zone of influence I lies. We assign $v(z) := w$, truncate the lower portion of I at z, thereby obtaining a new I^*, and, if the line passing through w and z has a slope greater than that of the edge of c_1 proceeding forward from z, add a new interval I_0 below I^*. All intervals of Q encountered during the ascending search for the interval containing z are deleted from Q. Similar and symmetric update rules are applied if z lies on c_2.

In this manner we obtain the values $v(z)$ for all $z \in N$. In processing the final point $B \in N$ no updating of Q is required; we only need to scan Q to find the interval containing B, and hence also $v(B)$.

Once the map v becomes available, the corners of the required shortest path p are (in reverse order) the points $x_1, \ldots, x_d$, where $x_1 = B$, $x_d = A$, and for each $i < d$ we have $x_{i+1} = v(x_i)$. This path is easily obtained by tracing the map v from B backwards until we reach A.

In regard to the running time of this algorithm, note that if an interval I' contained in the zone of influence of some $z \in N$ is deleted from $Q(t)$, then the zone of influence of z will never reappear in Q for any $t' > t$, and thus the number of deletions from Q is bounded by the total number n of corners in N. It is equally plain that the total number of truncations of intervals in Q, and the total number of insertions of new intervals into Q are both bounded by n. Since each deletion, truncation, and insertion can be performed in constant time, it follows that the shortest path algorithm that has just been described runs in time linear in n.

Remark: An interesting application of this smoothing procedure is as follows. Suppose that $y = c^*(x)$ is a noise-perturbed approximation to a smooth function, and that, given a sequence of points lying on the graph of c, we wish to find the regions of convexity and of concavity of c. Apply the above smoothing algorithm to the strip U bounded between $c_1(x) = c^*(x) - \epsilon$ and $c_2(x) = c^*(x) + \epsilon$, for sufficiently small ϵ. It can be shown that if the given sequence of points on c is sufficiently dense then good approximations to the regions of convexity can be characterized as regions in which the shortest path c' within U between the

endpoints of c passes through corners in the lower boundary c_1 of U, whereas regions of concavity will be very nearly those in which c' passes through corners of c_2. This method is preferable to direct comparison of the chord connecting two points y_1, y_2 (lying sufficiently close together on c) to their midpoint x on c, because c' eliminates much of the noise present in the raw data.

Smoothing an arbitrary simple polygonal plane curve.

The basic algorithm just described applies only to cases in which the curve c being approximated is monotonic in at least one coordinate direction. When this does not hold for the whole of c, a simple way of proceeding is to decompose c into disjoint sections, each of which is monotonic in some direction, and then to apply the above-described shortest path algorithm to each of these sections separately. In the experiments reported below, we employed this simple technique, and more precisely have used a decomposition scheme involving four directions of potential monotonicity - the x axis, the y axis, and the two lines at 45 degrees to these coordinate axes. Given a curve c, we find the longest initial portion of c which is monotonic in at least one of these four directions. After breaking off this initial section, we find the longest initial portion of the remainder of c which is monotonic in one of these directions, and continue in this manner until all of c is processed. This simple heuristic proves to be adequate even in the presence of relatively large amount of noise.

Remark: As noted above there exist algorithms due to Lee and Preparata [LP84] and to Chazelle [Ch82], for finding the shortest path between two specified points inside any given simple plane polygon. These algorithms run in time $O(n \log n)$, where n is the number of vertices of the polygon. A difficulty in applying these algorithms to the case of an arbitrary simple polygonal curve is that it is not clear what is the best way to define the polygon within which the shortest path will be sought. For this reason, simple extension of the linear-time algorithm given above for monotonic curves seems a suitable approach.

Approximation by nearly-shortest paths in three dimensions

Next we generalize the preceding considerations to curves in E^3. Since in this case no adequately efficient technique for finding shortest paths lying in polyhedral regions is known (cf. [SS84b], [Pa84]), we will use a modified shortening technique for smoothing which attains the efficiency required. As before, let c be a smooth curve, but now suppose that c is three-dimensional, and let c^* be obtained from c by addition of a perturbing (noise) function of maximum magnitude at most ϵ. For simplicity, we will suppose that c (resp. c^*) has a 1-1 parametrization of the form $(x,y(x),z(x))$ (resp. $(x,y^*(x),z^*(x))$). Define the neighborhoods $U_\epsilon(c)$ and $U_\epsilon(c^*)$ in terms of 'Manhattan' distance, e.g. $U_\epsilon(c^*)$ is the set of all points (x,y_0,z_0) such that x belongs to the (common) domain $0 \le x \le X$ of c and c^*, while

$$y^*(x) - \epsilon \le y_0 \le y^*(x) + \epsilon \ ,$$
$$z^*(x) - \epsilon \le z_0 \le z^*(x) + \epsilon \ .$$

The projection of $U_\epsilon(c^*)$ into the (x,y)-plane is the set $V_\epsilon(c^*)$ of all (x,y_0) satisfying $0 \le x \le X$ and $y^*(x) - \epsilon \le y_0 \le y^*(x) + \epsilon$. Define the function $y_1(x)$ for $0 \le x \le X$ by requiring that the curve $(x,y_1(x))$ shall be the shortest curve in $V_\epsilon(c^*)$ connecting the projections onto the (x,y)-plane A, B of the endpoints a, b of c^*. Since $V_\epsilon(c^*)$ is a 2-D 'band' of the form considered above, the planar shortest path algorithm that we have described calculates $y_1(x)$ efficiently.

Next let $W_\epsilon(c^*,y_1)$ be the set of all points $(x,y_1(x),z_0)$ such that $0 \le x \le X$ and $z^*(x) - \epsilon \le z_0 \le z^*(x) + \epsilon$, and define the function $z_1(x)$ for $0 \le x \le X$ by requiring that the curve $(x,y_1(x),z_1(x))$ shall be the shortest curve in $W_\epsilon(c^*,y_1)$ connecting the endpoints of c^*. Since $W_\epsilon(c^*,y_1)$ is not too different from the plane (it is in fact a 2-dimensional piecewise planar surface folded at those x which are the corners of $y_1(x)$) our planar shortest path algorithm also suffices to calculate $z_1(x)$ efficiently (simply apply this algorithm to the planar band obtained by unfolding $V_\epsilon(c^*,y_1)$). We shall show that, provided ϵ is small enough, the curve $c_1(x) = (x,y_1(x),z_1(x))$ is a good approximation to the original curve $c(x) = (x,y(x),z(x))$, in the sense that after both these curves are parametrized by arc-length

they must approximate each other uniformly.

For this we can employ much the same argument used to prove Lemma 2.2, provided that we first show that the curve c_1 is not much longer than the model curve c. Note that since c_1 lies within $U_{2\epsilon}(c)$, the length of c_1 must be at least $L - \delta$, since we mean to assume that c has length L and is $(2\epsilon,\delta)$-stable.

To prove that c_1 is not much longer than c, we argue as follows. In a neighborhood O of the projected planar curve $C = (x,y(x))$ introduce local curvilinear coordinates (s,u) which map C onto the x-axis, s being arc-length along the curve C, and which map each sufficiently short segment orthogonal to the curve C into a segment of the form (s_0,u), $-a(s_0) \le u \le b(s_0)$. (Here it is plain that a and b must be positive functions.) In these curvilinear coordinates the Euclidean (x,y)-length of any plane curve $p(t)$ is expressed by an integral

$$\int (p'(t)\, G(p(t))\, p'(t)^T)^{1/2} dt\,, \tag{1}$$

where as usual $p'(t)$ designates the derivative of the (vector-valued) function $p(t)$, and where $G(p)$ is a positive definite 2×2 matrix. Moreover, since arc-length along the s-axis is measured directly by the coordinate s, and since the lines (s_0,u) are all orthogonal to the s-axis in the metric defined by G, we must have $G((s,0)) = I$ identically in s, where I designates the 2×2 identity matrix. It follows that for each positive constant η we have

$$v\, G((s,u))\, v^T \ge (1-\eta)^2 v\, v^T$$

for all vectors v and all sufficiently small u.

If ϵ is sufficiently small, the minimum length curve $(x,y_1(x))$ in $V_\epsilon(c^*)$ introduced above can be written in (s,u) coordinates as $(s,Y(s))$ where, unfortunately, the function Y may be multi-valued. (However, by perturbing $y_1(x)$ very slightly, we can assume that only a finite number of smooth 'branches' of $Y(s)$ lie over each interval of the s-axis.) This will cause slight technical complications but will be seen not to disturb the overall course of the argument we are about to give. Let L_C be the length of the curve C. Since C is included in the projected neighborhood $V_\epsilon(c^*)$ of c^*, the length of the curve $(x,y_1(x))$ is no more than

L_C. Expressing this fact in the curvilinear coordinates (s,u), we find that

$$\int_0^{L_C} \left((1,Y'(s))\, G((s,Y(s)))\, (1,Y'(s))^T \right)^{1/2} ds \le L_C \,. \tag{2a}$$

Hence, assuming that ϵ is sufficiently small relative to the positive constant η chosen above, we have

$$(1-\eta) \int_0^{L_C} (1 + Y'(s)^2)^{1/2}\, ds \le L_C \,, \tag{2b}$$

i.e.

$$\int_0^{L_C} \left[(1 + Y'(s)^2)^{1/2} - 1 \right] ds \le \frac{\eta}{1-\eta} L_C \,. \tag{2b}$$

Note that the inequalities (2a) and (2b) hold exactly as written if the function $Y(s)$ is single valued, whereas if $Y(s)$ is multi valued and represented by multiple branches over certain intervals of the s-axis, the (positive) integral in (2a) and the integral in (2b) should be understood to represent the sum of corresponding single valued integrals, the sum being extended over all the branches of $Y(s)$.

It is easy to see that the following inequalities hold:

$$(1 + a^2)^{1/2} - 1 \ge \frac{1}{3} a^2 \,, \qquad 0 \le a \le 1 \,,$$

$$(1 + a^2)^{1/2} - 1 \ge \frac{1}{3} a \,, \qquad 1 \le a \,.$$

It therefore follows from (2b) that

$$\int_{|Y'(s)| \le 1} |Y'(s)|^2\, ds \le 3 \frac{\eta}{1-\eta} L_C \tag{3a}$$

and

$$\int_{|Y'(s)| > 1} |Y'(s)|\, ds \le 3 \frac{\eta}{1-\eta} L_C \,. \tag{3b}$$

As before, these inequalities are valid exactly as written if the function $Y(s)$ is single valued, whereas if $Y(s)$ is multi valued and represented by multiple branches over certain intervals of the s-axis, each of the integrals in (3a) and (3b) should be understood to represent the sum of corresponding single valued integrals, the sum being extended over appropriate portions of

all the branches of $Y(s)$.

Our next step is to show that the curve $c_3(x) = (x, y_1(x), z(x))$, $0 \le x \le X$, is not much longer than the curve $c(x) = (x, y(x), z(x))$, i.e. has length not much greater than L. To see this, parametrize c_3 in terms of s as $(s, Y(s), Z(s))$, where just as above the function Y is multi-valued and can therefore cause slight technical complications. (However, $Z(s)$ is single-valued since it merely represents $z(x)$ which is single-valued along the curve c, i.e. the s-axis.)

The positive-definite matrix G appearing in (1) satisfies $G((s,u)) \le (1+\eta)\, I$ for all sufficiently small u. Thus, assuming as above that ϵ is sufficiently small, we have

$$\int_0^{L_C} \left((1, Y'(s))\, G((s, Y(s)))\, (1, Y'(s))^T + Z'(s)^2 \right)^{1/2} ds \tag{4}$$

$$\le (1+\eta) \int_0^{L_C} \left(1 + Y'(s)^2 + Z'(s)^2 \right)^{1/2} ds \, ,$$

where as above we may be required to sum in appropriate fashion over multiple branches of $Y(s)$. The right hand side of the inequality (4) can be written as

$$(1+\eta) \left(\int_0^{L_C} (1 + Z'(s)^2)^{1/2}\, ds + \int_0^{L_C} \left[(1 + \frac{Y'(s)^2}{1 + Z'(s)^2})^{1/2} - 1 \right] (1 + Z'(s)^2)^{1/2}\, ds \right) \tag{5}$$

and the second integral B in (5) satisfies

$$B \le \frac{1}{2} \int_{|Y'(s)| \le 1} |Y'(s)|^2\, ds + \int_{|Y'(s)| > 1} |Y'(s)|\, ds \le 5 \frac{\eta}{1-\eta} L, \tag{6}$$

by (3a) and (3b), since $(1 + a^2)^{1/2} - 1 \le \frac{1}{2} a^2$ and $(1 + a^2)^{1/2} - 1 \le a$ for all $a \ge 0$. As before, let L be the length of the model curve $c = (x, y(x), z(x))$, so that plainly $L_C \le L$ Since the first integral in (5) (resp. the integral in the left-hand side of (4)) is simply L_C (resp. the length L_3 of c_3, we find that

$$L_3 \le (1+\eta)\, L + 5 \frac{\eta\,(1+\eta)}{1-\eta}\, L \, , \tag{7}$$

showing, as required, that when ϵ is sufficiently small, L_3 is not much larger than L.

Since the curve $c_3(x) = (x, y_1(x), z(x))$ belongs to the set $W_\epsilon(c^*, y_1)$ introduced above,

and since $c_1(x) = (x, y_1(x), z_1(x))$ is the shortest curve in $W_\epsilon(c^*, y_1)$ connecting endpoints which differ only slightly from those of c_3, it follows that c_1 also has length differing only slightly from that of c. Thus, as noted above, c_1 will be uniformly good approximation to c if ϵ is sufficiently small and both c and c_1 are parametrized by their arc length.

The preceding discussion assumes that the whole of the given curve c can be parametrized by the coordinate x. To treat more general simple 3-D curves we can decompose them into monotonic subsections using much the same heuristic described for 2-D curves, and then apply the shortening procedure just described to each one of these sections.

Fig. 2.1 and Fig. 2.2 show a 2-D curve artificially perturbed by addition of random noise, and the result of smoothing this curve by the 'shortening' method just described; figs. 4.1 and 4.2 of section 4 do the same for a 3-D curve.

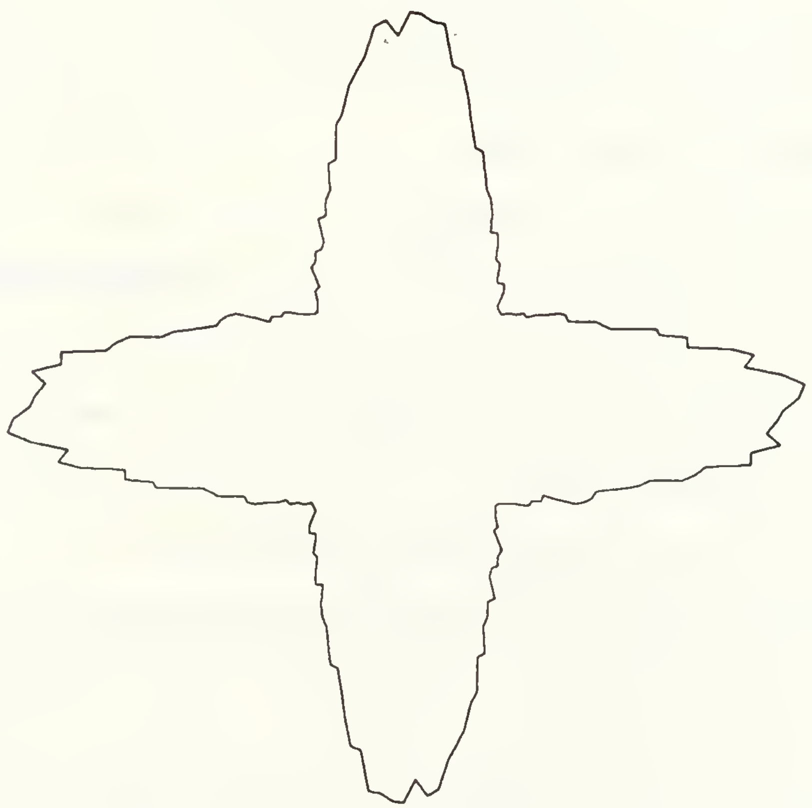


Fig. 2.1. An Artificially Perturbed 2-D Curve.

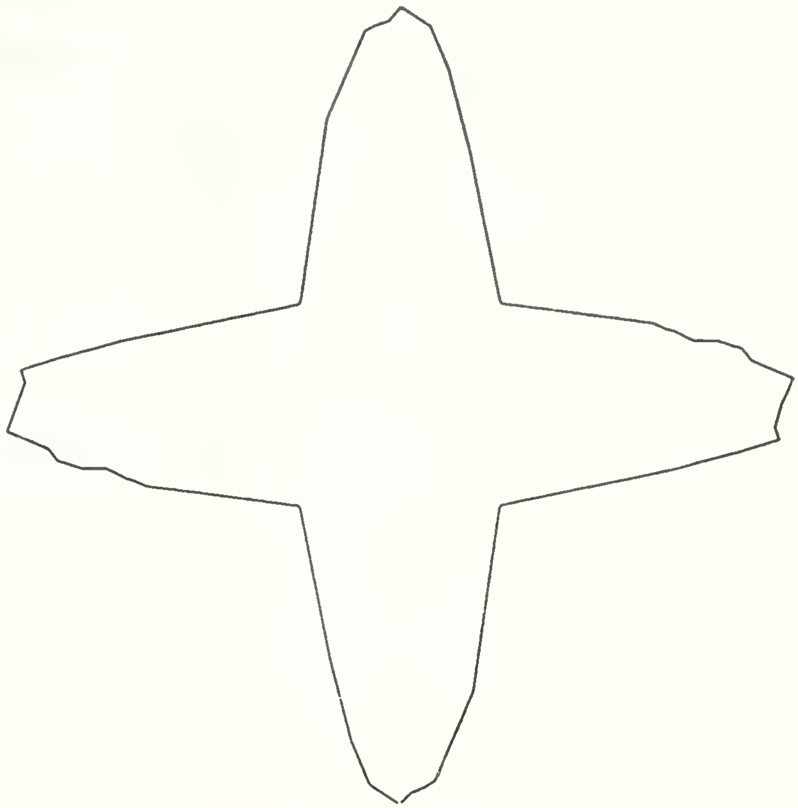


Fig. 2.2. Fig. 2.1 Curve After Smoothing.

3. Fast boundary matching by least squares fitting.

Next we describe a fast technique for matching a portion of the boundary C' of an observed image O' against a corresponding portion of the boundary C of a model object O. We assume that C' has been smoothed using one of the smoothing methods described in the preceding section, and that both C and C' are parametrized by arc length. Recall that the matching we seek calls for determination of the offset s_0 and the Euclidean transformation E for which the curves $EC(s+s_0)$ and $C'(s)$ are closest to one another in the L_2 norm. To be more specific, represent each of the curves C, C' by a sequence of evenly spaced points on it, and let these sequences be $(\mathbf{u}_j)_{j=1}^{n}$ and $(\mathbf{v}_j)_{j=1}^{n}$, corresponding to an observed (smoothed) object O' and a model object O respectively. For simplicity in describing the necessary calculation, assume first that the whole boundary of O' is visible and that no offset calculation is required. Matching then amounts to finding the Euclidean motion E of the plane which minimizes the l^2 distance between the sequences $(E\mathbf{u}_j)_{j=1}^{n}$ and $(\mathbf{v}_j)_{j=1}^{n}$; i.e. we

need to compute

$$\Delta = \min_E \sum_{j=1}^{n} |E\mathbf{u}_j - \mathbf{v}_j|^2$$

To simplify the calculation, we first translate O' so that the centroid of its boundary lies at the origin, i.e. so that

$$\sum_{j=1}^{n} \mathbf{u}_j = 0$$

Next we write E as $E\mathbf{u} = R_\theta \mathbf{u} + \mathbf{a}$, R_θ denoting a counterclockwise rotation by θ. Then

$$\Delta = \min_{\theta,\mathbf{a}} \sum_{j=1}^{n} |R_\theta \mathbf{u}_j + \mathbf{a} - \mathbf{v}_j|^2 =$$

$$\min_{\theta,\mathbf{a}} \left[\sum_{j=1}^{n} |\mathbf{v}_j|^2 + n|\mathbf{a}|^2 - 2\sum_{j=1}^{n} \mathbf{a}\cdot\mathbf{v}_j + \sum_{j=1}^{n} |\mathbf{u}_j|^2 + 2\sum_{j=1}^{n} \mathbf{a}\cdot R_\theta \mathbf{u}_j - 2\sum_{j=1}^{n} R_\theta \mathbf{u}_j \cdot \mathbf{v}_j \right]$$

But

$$\sum \mathbf{a}\cdot R_\theta \mathbf{u}_j = \mathbf{a}\cdot R_\theta (\sum \mathbf{u}_j) = 0 .$$

Hence $\mathbf{a}$ and θ appear independently in Δ and we can minimize their contributions separately.

To minimize over $\mathbf{a}$ we simply put

$$\mathbf{a} = \frac{1}{n} \sum_{j=1}^{n} \mathbf{v}_j$$

As to θ, we need to compute

$$\delta = \max_\theta \sum_{j=1}^{n} R_\theta \mathbf{u}_j \cdot \mathbf{v}_j$$

Regarding the vectors $\mathbf{u}_j$, $\mathbf{v}_j$ as complex numbers u_j, v_j, we can rewrite this as

$$\delta = \max_\theta \ \mathrm{Re} \left[\sum_{j=1}^{n} e^{i\theta} u_j \bar{v}_j \right] = |\sum_{j=1}^{n} u_j \bar{v}_j| ,$$

where plainly the maximizing θ is just the negative of the polar angle of $\sum u_j \bar{v}_j$. Altogether this gives

$$\Delta = \sum_{j=1}^{n} |\mathbf{v}_j|^2 - \frac{1}{n} |\sum_{j=1}^{n} \mathbf{v}_j|^2 + \sum_{j=1}^{n} |\mathbf{u}_j|^2 - 2|\sum_{j=1}^{n} u_j \bar{v}_j| \qquad (*)$$

$$\Delta = \sum_{j=1}^{n} |\mathbf{v}_j|^2 - \frac{1}{n} |\sum_{j=1}^{n} \mathbf{v}_j|^2 + \sum_{j=1}^{n} |\mathbf{u}_j|^2 - 2\left(|\sum_{j=1}^{n} \mathbf{u}_j \cdot \mathbf{v}_j|^2 + |\sum_{j=1}^{n} \mathbf{u}_j \times \mathbf{v}_j|^2 \right)^{\frac{1}{2}} ,$$

where $\mathbf{u}\times\mathbf{v}$ denotes the (2-dimensional) cross product of the vectors $\mathbf{u}$ and $\mathbf{v}$.

If O' is partially occluded, we have to match the sequence $(\mathbf{u}_j)_{j=1}^{n}$ to each of the contiguous subsequences $(\mathbf{v}_{j+d})_{j=1}^{n}$ of the (circular) sequence $(\mathbf{v}_j)_{j=1}^{m}$, for $d = 0, \ldots, m-1$. (Here we can appropriately assume that $m \geq n$, for otherwise the (partial) periphery of O' is too long to match O.)

For each such d (*) becomes

$$\Delta(d) = \sum_{j=d+1}^{d+n} |\mathbf{v}_j|^2 - \frac{1}{n}\left|\sum_{j=d+1}^{d+n} \mathbf{v}_j\right|^2 + \sum_{j=1}^{n} |\mathbf{u}_j|^2 - 2\left|\sum_{j=1}^{n} u_j \bar{v}_{j+d}\right|$$

It is easily seen that the minimum of the values $\Delta(d)$, $d = 0, \ldots, m-1$, can be found in time $O(m \log m)$, by using the fast Fourier transform algorithm for computing the convolutions $\sum_{j=1}^{n} u_j \bar{v}_{j+d}$.

The matching technique just described generalizes easily to curves in three dimensions, or, more generally, to any situation in which a model curve or surface $\mathbf{u}(\omega)$ depending on one or more parameters ω must be rotated and translated to match a model curve or surface $\mathbf{v}(\omega)$ as well as possible. We must assume, however, that the matching operation involves no change in parametrization for either of the functions $\mathbf{u}(\omega)$ or $\mathbf{v}(\omega)$.

Suppose then that we are given two descriptor functions $\mathbf{u}(\omega)$, $\mathbf{v}(\omega)$, $\omega \in S$, corresponding respectively to an observed curve or surface C' and a model curve or surface C. We need to find the Euclidean motion E (e.g. of 3-space) which minimizes

$$\Delta = \min_E \int_S |E\mathbf{u}(\omega) - \mathbf{v}(\omega)|^2 \, d\omega$$

(The domain S of integration can have one or more dimensions.) As in the 2-D case, we translate C' so that its centroid lies at the origin, giving

$$\int_S \mathbf{u}(\omega) \, d\omega = 0$$

Write E as $E\mathbf{u} = R\mathbf{u} + \mathbf{a}$, where R is a rotation. Then

$$\Delta = \min_{R,\mathbf{a}} \int_S |R\mathbf{u}(\omega) + \mathbf{a} - \mathbf{v}(\omega)|^2 \, d\omega =$$

$$\min_{\mathbf{a}} \left[\int_S |\mathbf{v}(\omega)|^2 \, d\omega + |S||\mathbf{a}|^2 - 2\int_S \mathbf{a}\cdot\mathbf{v}(\omega) \, d\omega + \int_S |\mathbf{u}(\omega)|^2 \, d\omega + 2\int_S \mathbf{a}\cdot R\mathbf{u}(\omega) \, d\omega - 2\int_S R\mathbf{u}(\omega)\cdot\mathbf{v}(\omega) \, d\omega \right]$$

But

$$\int_S \mathbf{a}\cdot R\mathbf{u}(\omega)\, d\omega = \mathbf{a}\cdot R(\int_S \mathbf{u}(\omega)\, d\omega\,) = 0\,.$$

Hence $\mathbf{a}$ and R appear independently in Δ and we can minimize their contributions separately.

To minimize over $\mathbf{a}$ we set

$$\mathbf{a} = \frac{1}{|S|}\int_S \mathbf{v}(\omega)\, d\omega$$

As to R, we need to compute

$$\delta = \max_R \int_S R\mathbf{u}(\omega)\cdot\mathbf{v}(\omega)\, d\omega$$

To find δ we first calculate the matrix A given by

$$A_{ij} = \int_S \mathbf{u}_i(\omega)\mathbf{v}_j(\omega)\, d\omega\ ,$$

(where $i,j = 1,2,3$ if we are dealing with a curve or surface in 3-space). In terms of the matrix A we can express δ as

$$\delta = \max_R \ tr(RA)$$

To maximize $tr(RA)$ first assume A to be non-singular; then we can decompose A as $A = QH$, where $Q = A(A^*A)^{-\frac{1}{2}}$ is a pure rotation and $H = (A^*A)^{\frac{1}{2}}$ is positive definite symmetric. This gives

$$\delta = \max_R \ tr(RQH) = \max_R \ tr(RH) = tr(H) = tr((A^*A)^{\frac{1}{2}})$$

Indeed, since the trace is invariant under rotation, we may assume that H is diagonal. But since for a diagonal positive definite matrix (λ_i) and a rotation matrix (r_{ij}) the trace of their product $\sum \lambda_i r_{ii}$ can be no larger than $\sum \lambda_i$ and can assume this value only when $r_{ij} = \delta_{ij}$. An identical result for a singular A follows easily by continuity arguments. Altogether we have

$$\Delta = \int_S |\mathbf{v}(\omega)|^2\, d\omega - \frac{1}{|S|}|\int_S \mathbf{v}(\omega)\, d\omega|^2 + \int_S |\mathbf{u}(\omega)|^2\, d\omega - 2\, tr((A^*A)^{\frac{1}{2}}) \qquad (^{**})$$

showing that the optimal rotated match between C' and C can be found in time proportional to that needed to integrate the various functions appearing in (**), i.e. proportional to the number of data points used to discretize the curves or surfaces **u** and **v**. In particular, as in the 2-D case considered previously, these formulae can be used to match observed 3-D curves parametrized by arc length to similarly parametrized model curves. Matching can be achieved in time $O(n \log n)$ by using the fast Fourier transform, even if the observed curve O' is partially obscured. Formula (**) gives us the smallest least-squares distance between the objects C and C', but does not produce the Euclidean transformation E which attains this best match. In case A is nonsingular, the required E is given by $EU = R\ u + a$ where a is as given above, and $R = (A^*A)^{1/2} A^{-1}$. Even if A is singular, it can be factored as $A = QH$ where Q is a rotation and H is as above, but in this case any rotation of the form $R = R_0 Q^{-1}$, where R_0 is the identity on the range of H but can be arbitrary on the space orthogonal to this range, will minimize $tr(RA)$. The map Q can be constructed in this singular case by finding a maximal linearly independent set of columns of H, and then forming Q which maps each such column into the corresponding column of A and which is the identity on each element of the (common) nullspace of A and H.

4. Application to the analysis of compound 2-D scenes

A 'compound' 2-D scene is one consisting of thin objects, e.g. objects cut out of metal or cardboard, which may overlap. To apply a boundary matching procedure like that described in the preceding section to such scenes, one needs a way of determining where the boundary of one of several overlapping objects ends and that of another begins. For obvious reasons, these 'breakpoints' are likely to be points at which the tangent to the overall boundary B of the overlapping objects changes rapidly, hence points at which the second derivative of the boundary curve reaches a peak. To be more specific, let $b(s)$ be a parametrization of the (smooth) boundary B by arc length, and suppose that $b(s)$ has the additional property that as we traverse B in the direction of increasing s, the interior of the

(union of the) overlapping objects lies to the right of B. Let $n(s)$ be the outward-pointing unit normal to B at $b(s)$. Then the second derivative $b''(s)$ is parallel to $n(s)$, and we look for points s where $b_2(s) = b''(s) \cdot n(s)$ reaches a positive peak. (Only positive peaks are of concern since negative peaks correspond to sharp boundary convexities rather than concavities, which are not likely to be points where the boundaries of two distinct overlapping objects meet.)

Specifically, assume that the boundary B has been smoothed and that we have enumerated a sequence of evenly spaced points in some plausible clockwise order around B. Then we can proceed in the following manner.

(a) Choose some fixed (and relatively small) k in advance, take k successive boundary points starting at each successive place j, and estimate the boundary tangent by calculating the line of best (least squares) fit to these successive points.

(b) Calculate the second derivative by differencing successive tangents (see below for details), and look for maxima of the second derivative. (As noted, by ignoring minima we bypass convex corners and record only concave corners.)

(c) If many maxima lie close to each other, thin them out.

The necessary calculations are fast and simple. Let the k points presently being processed be $p_i = (x_i, y_i)$, $i = 1, \ldots, k$. The horizontal line of best approximation is at the level y which minimizes

$$\frac{1}{k} \sum_{1}^{k} (y_i - y)^2 ,$$

hence $y = \frac{1}{k} \sum y_i$, and the minimum is

$$\frac{1}{k} \sum y_i^2 - (\frac{1}{k} \sum y_i)^2 .$$

This analysis applies to lines of any orientation, and thus the line of best approximation must pass through the center of gravity of the set of points p_i, and the unit normal v to this line must be that which minimizes

$$H(v) = k \sum (v \cdot p_l)^2 - (v \cdot \sum p_l)^2 = v\,(P - Q \circ Q)\,v\,,$$

where P is the dyadic sum $\sum p_l \circ p_l$, and $Q = \sum p_l$. The minimizing vector is therefore the eigenvector belonging to the smaller eigenvalue of the symmetric 2×2 matrix $A = P - Q \circ Q$, which we can write as

$$\begin{pmatrix} a & b \\ b & c \end{pmatrix},$$

whose characteristic equation is $\lambda^2 - (a+c)\lambda + ac - b^2$, giving the smaller eigenvalue

$$\lambda = \frac{a + c - \left((a - c)^2 + 4b^2\right)^{1/2}}{2},$$

and a corresponding eigenvector proportional to both $(-b,\, a-\lambda)$ and $(c-\lambda,\, -b)$. (Care has to be taken to use the larger of these two vectors, so as to avoid situations in which the smaller of the two vanishes.)

The calculations required can therefore be pertormed quite efficiently by a linear scan through the sequence of boundary points.

To find concave corners using these best-fitting tangents we can take the cross product of each such vector τ with the next tangent, which gives an estimate for the angle between them. Then we can look for positive peaks in this angle at which it exceeds some threshold angle $\frac{\phi_0}{k}$, the denominator k being appropriate since least square fitting averages the tangent slope among k successive points, making it reasonable to expect that roughly k steps will transpire before one sees all of any fast change in tangent orientation.

Simulations in 2 and 3 dimensions; Experiments with real 2-D images

This technique just explained has been simulated in combination with the smoothing method described above. It appears to work well and to withstand considerable noise in the boundary curves simulated. Figures 4.1 and 4.2 show the result of one such simulation. Fig . 4.1 figure shows (a 3-D) a test curve and the result of perturbing a translated and rotated subsection of it by artificial noise. Fig. 4.2 shows the matching of the noisy figure to the original curve obtained after smoothing using the 3-D smoothing procedure described above.

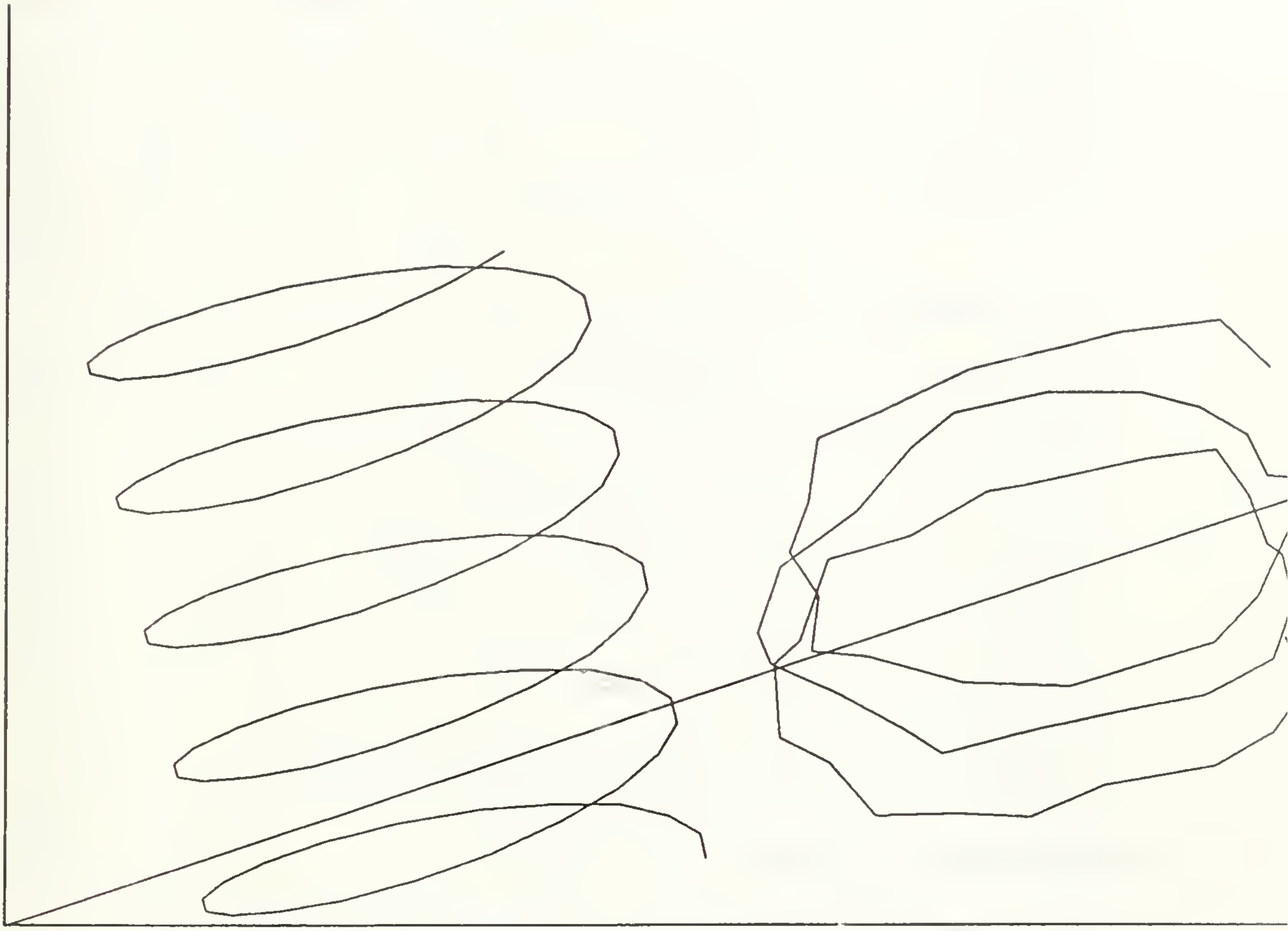


Fig. 4.1. A Test Curve And A Perturbed, Rotated And Translated Subsection.

The companion paper [KSSS85] reports on various much more extensive experiments with partially obscured 2-D curves and composites of such curves, which show that the techniques described above can be used to analyze such configurations robustly and reliably.

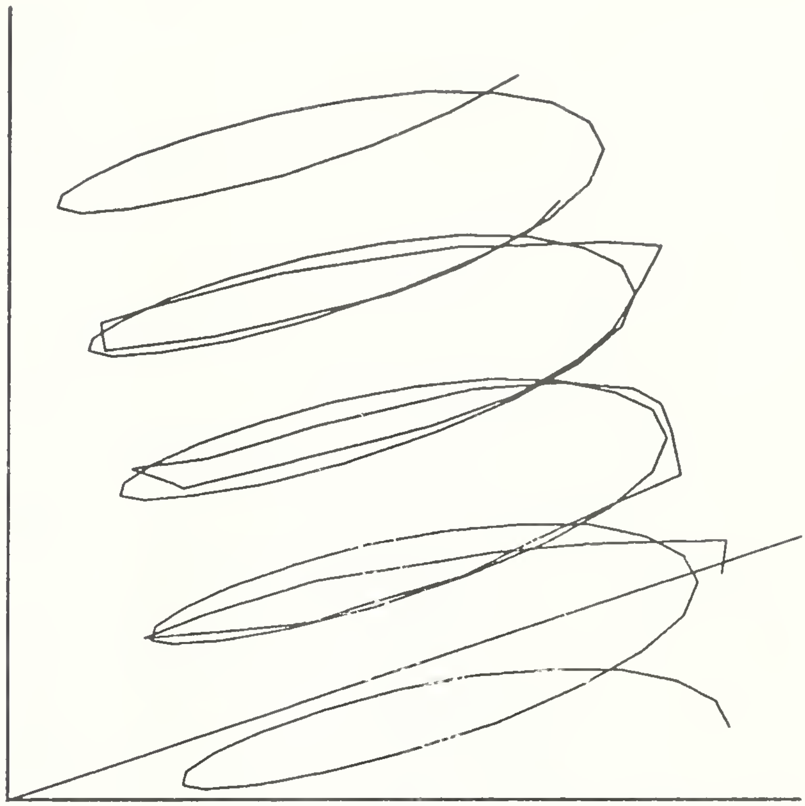


Fig. 4.2. Fig. 4.1 Curve Smoothed And Matched To Model Curve.

5. Characteristic Curves on the Visible Surfaces of 3-D Bodies

To apply the matching method described in the preceding pages to partially obscured 3-D bodies O, the most desirable approach is to define and match suitable characteristic curves on their surfaces. (If such curves are not available, we will have to match observed to model 2-D surfaces, which may be a much more expensive operation, especially if we do not know which portion of a 3-D model object has been observed.) Surface curves used for recognition must be curves C defined by the geometry or surface color of a surface O and fixed in O, so that when O undergoes a rotation and/or translation the curve C moves along with it. C's definition should depend only on local geometric and other features of O, so that parts of C can be constructed even when only a portion of O's surface can be observed. Such characteristic curves can be defined in any one of the following ways:

(i) **Use of albedo or color variations.** Suppose that the surface of O is 'painted', e.g. that portions of this surface have sensibly different reflectances in regions of the electromagnetic

spectrum. A simple example might be a white sphere or other convex body with red curves painted on it. Then any sharply defined reflectance transitions can be used as characteristic curves. Moreover, even if no sharp transitions exist because O's reflectance varies smoothly over its surface, it may still be possible to use the 'level' curves along which this reflectance has a given value as characteristic curves.

Note that we assume here that a 3-D or 'depth' image of the visible surfaces of O is available, so that the true 3-D geometry of each characteristic curve C observed on O's surface is known. Such information can easily be obtained by combining any suitable depth sensor with a standard video camera sensitive to appropriate parts of the optical spectrum. (When color variations on O's surface are being used, it may be advantageous to eliminate Lambert-law dependencies of reflectance on the angles of illumination and observation by forming ratios of two or more separate images acquired using light of various wavelengths.)

(ii) **'Curves of rolling'.** Next consider the somewhat more difficult case in which O's surface is 'pure white', i.e. in which only geometric information, but no useful reflectance information, is available concerning these surfaces. Even in this case it is not hard to define characteristic curves. The easiest case to consider is that in which O is not convex, in which case 'curves of rolling' on O's surface will exist and can be used as characteristics. These curves are defined as follows. Each tangent plane to a smooth convex object O touches it in just one point, but if O is not convex it will have tangent planes P which touch it in at least two separated points. (We ignore the even more favorable case of tangent planes which touch O's surface in three separated points, since in general there can exist at most a finite number of such planes, and hence as soon as one such plane has been found the problem of identifying O and determining its orientation becomes purely discrete (This remark applies to polyhedra, and makes them easy to identify by a relatively straightforward 'probing' method see [GLP84], [GLP85].)

We shall say that a point p on the surface of a smooth object O is a *point of rolling* if the tangent plane to O at p is also tangent to O at another point q, and call the locus of all such

points p the *curves of rolling* on the surface of O. (This name is suggested by the fact that, given a physical model of O, curves of rolling on it can be found by putting O on a flat tabletop T and finding positions in which O rolls on T with two points in contact with T. The points at which O contacts T as it rolls along the one-parameter path available to it are then curves of rolling.)

To see that the locus of points of rolling on a smooth 3-D body O can generally be expected to lie along curves, we can argue as follows. Map each point p of O's surface S into the plane $P(p)$ tangent to O at p. This maps the two-dimensional surface S', into the space Π of all planes in E^3. A plane is defined by its unit normal vector and distance from the origin, three parameters in all, so that Π is a 3-dimensional manifold into which P maps S. The points of rolling correspond to points of self-intersection of the image surface $P(S)$, and thus since self-intersections of a 2-dimensional surface embedded (in general position) in a 3-dimensional space like Π must lie along 1-dimensional curves, it follows that the points of rolling on O's surface will generally lie along smooth curves.

Note that points of rolling can be found as soon as enough of O's surface to show the presence of two convexities separated by a concavity has been observed.

(iii) **Other geometrically defined invariants.** Even if O is convex (or if its concavities are not observed), other geometric invariants of its surface can be used to define characteristic curves. Generally speaking, any function $f(p)$ defined for points p on the surface of O and rotating with O will define characteristic 'level curves' $f(p) = k$. Another possibility is to use a function of this kind which is defined on some auxiliary surface O' associated with O that rotates with O. For example, we can let O' be the sphere of unit vectors v, and for each $d > 0$ and $v \in O'$ define $f_d(v)$ as follows: take a plane perpendicular to v, initially very far out in the direction of v, and then translate this plane P back along v (maintaining its orientation) until it first touches O and then penetrates into O to depth d. Let $f_d(v)$ be some rotational invariant of the intersection of P with O, e.g. the area, central moment, or perimeter of this intersection. Another possibility (but one more apt to be

sensitive to noise) is to map each point p of O to its normal vector $N(p)$, and to define the characteristic function f_d by some approximation to the Jacobian of the map N (e.g. the area on the unit sphere swept out by $N(q)$ as q traverses the part of O's surface which lies within distance d of p).

6. A surface-based matching procedure for recognition of a general class of 'machined' objects.

Having discussed curve-based matching procedures in the preceding sections, we now go on to consider what can be done by way of matching surfaces on which stably defined curves cannot easily be found. Here we will only be able to deal efficiently with relatively simple surfaces. Fortunately, however, the surfaces of machined objects are often quadric, or, if this is not the case, can be represented well by quadratic or cubic splines. The following paragraphs will describe a matching technique for such surfaces, which can be made particularly effective for quadric surfaces. The technique in question is applicable to any surface having an implicit polynomial representation $P(u,v,w)=0$.

Suppose that a finite collection of model surfaces with such descriptions is given, that a collection of points $x_l = (u_l, v_l, z_l)$ on one of these surfaces is observed, but that the Euclidean orientation E of the observed surface is unknown. We can define the 'degree of match' between the set of observations and the equation $P(x)=0$ for the model as the minimized difference

$$\min_E \sum w_l \left(P(Ex_l)\right)^2, \tag{1}$$

where E ranges over the group of Euclidean transformations, and where the w_l are weights chosen to reflect some heuristic notion of the 'evidential weight' of each observation. (For example, w_l can be the distance, or the squared distance, of the observed point x_l from the average of all observations; this will increase the 'evidential weight' assigned to outlying points, which may be appropriate).

A difficulty is that (1) involves a nonlinear minimization leading to equations which will

generally be of unpleasantly high degree. To sidestep this difficulty, we can minimize, not over the map E, but over the coefficients of all polynomials of the same degree as the model surface $P(x)=0$ we are trying to match. That is, we can find the polynomial Q, having the same degree as P, at which the minimum

$$\min_{Q} \sum w_l \left(Q(x_l)\right)^2, \qquad (1')$$

is attained. This is a much more tractable quadratic minimization problem. Moreover, if the observed points did lie exactly on the surface $P(Ex)=0$, then the minimizing Q in (1') would clearly be $Q(x)=P(Ex)$, so that once having the coefficients of Q we could expect to calculate E in straightforward fashion. (Suppose, for example, that the polynomial P is of even order. The rotational part R of E acts independently of E's translational part on the highest order terms of P. If P is quadratic, R can be determined from Q by finding the principal axes of Q's purely quadratic part and matching them with the corresponding axes for P. If P is of higher degree, we can apply the Laplacian operator Δ, which is rotationally invariant, to the highest order terms of both P and Q just often enough to produce two quadratic polynomials, and then match principal axes as before. Next suppose that the principal terms of P are cubic. In this case, we can associate an orthogonal set of 'principal axes' with P in the following way. Take ΔP, which is linear, and take the vector v_l orthogonal to the plane $\Delta P=0$. This is the first principal axis of P. Then take the (cubic) restriction of P to the plane $\Delta P=0$, and repeat this procedure recursively to define $n-1$ additional axes $v_2,\ldots,v_n$ which are orthogonal to v_1. These axes have an invariant relationship to P, in the sense that they simply rotate when P is transformed into $P(Rx)$. If Q and P are both cubic, they can be matched simply by matching their characteristic axes; if they are of higher degree, we can apply the Laplacian operator an appropriate number of times to both Q and P, and then match the resulting cubic polynomials. Moreover, once the relative rotational positions of P and Q have been estimated, their relative translational position can be estimated even more easily by matching the terms of next-to-highest degree of P to those of Q.

Each of the coefficients of the polynomial Q appears quadratically in the sum (1') to be minimized. If the rotated polynomial Q is known to belong to some linear subspace of polynomials not necessarily passing through the origin (or, equivalently, if P is known to belong to some rotationally invariant linear subspace of polynomials) then it is appropriate to minimize over this space only. Similarly if P is known to belong to some rotationally invariant quadratic surface in the space of polynomials, it is appropriate to minimize for Q satisfying this constraint also. Beyond this, imposition of more complex constraints, or of more than one quadratic constraint, significantly increases the cost of minimization.

If the polynomial P is quadratic it will satisfy a condition $\Delta P = c$ for some constant c, and since this linear condition is rotationally invariant we can subject Q to it also. Moreover, since in this case $P_l = \frac{\partial}{\partial x_l} P$ defines a vector of polynomials, the polynomial $B = \sum P_l^2$ is also of second order and will satisfy a second condition $\Delta B = c_2$, which is quadratic in the coefficients of the the original polynomial P. Hence we can assume that Q satisfies this same quadratic condition. It follows that an appropriate way of matching a given model polynomial P is to find the Q which minimizes

$$\min_{\substack{\Delta Q = c_1 \\ \Delta \sum \left(\frac{\partial}{\partial x_l} Q\right)^2 = c_2}} \quad \sum w_l \left(Q(x_l)\right)^2, \tag{1''}$$

where the coefficients c_1 and c_2 are taken from the model polynomial P. Note that if $Q(x) = xAx + v \cdot x + b$ is a quadratic polynomial, then $\Delta Q = 2tr\,(A)$ and $\sum \left(\frac{\partial}{\partial x_l} Q\right)^2 = (2Ax + v) \cdot (2Ax + v) = 4x\,A^2 x + 4Ax \cdot v + v^2$, so $\Delta \sum \left(\frac{\partial}{\partial x_l} Q\right)^2 = 8\,tr\left(A^2\right)$.

The constraints appearing in the minimum (1) therefore restrict the $\sum a_{ll}$ and $\sum a_{lj}^2$, where $A = \left(a_{lj}\right)$ is the symmetric matrix of the quadratic (i.e. highest order) terms in the polynomial Q.

In somewhat more detail: The sum $\sum w_l \left(Q(x_l)\right)^2$ appearing in (1'') can easily be expressed in terms of the coefficients of the polynomial

$$Q(x) = \sum C_{k\,l\,m}\, u^k\, v^\ell\, z^m,$$

where $x = (u, v, z)$ namely

$$\sum w_l\,(Q(x_l))^2 = \sum C_{\bar{k}\bar{\ell}\bar{m}} \left(\sum_l w_l\, u_l^{k+\bar{k}}\, v_l^{\ell+\bar{\ell}}\, w_l^{m+\bar{m}}\right) \qquad \mathrm{I}$$

$$= \sum C_{k\ell m}\, C_{\bar{k}\bar{\ell}\bar{m}}\, K_{k\ell m,\,\bar{k}\bar{\ell}\bar{m}} \qquad \mathrm{I}$$

where the symmetric matrix K is positive definite. (Note that each of the entries of this matrix is simply a particular weighted moment of the collection of surface points observed, this moment having a degree which is at most twice the degree of the polynomial equation $Q(x) = 0$ which is to be matched to these surface points.) In particular, if $Q(x) = x\,A\,x\; + b\,x\; + c$ is a quadratic polynomial, and if we group the three components of b together with the scalar c to form a 4-dimensional quantity $B = (b, c)$, we can write the quadratic form (2) as

$$AK_1\,A\; + \;2BK_2\,A\; + \;BK_3\,B, \qquad (2')$$

and so can rewrite the minimum (1'') in the form

$$\min_{\substack{tr(A)=c_1\\ tr(A^2)=c_2}}\ \min_B\ (AK_1A+2BK_2A+BK_3B) \qquad (2'')$$

$$= \min_{\substack{tr(A)=c_1\\ tr(A^2)=c_2}}\ A(K_1-K_2^*K_3^{-1}K_2)A.$$

We can then write $A=\frac{1}{3}c_1I+A_0$, where $tr(A_0) = 0$, allowing the minimum (2") to be rewritten as

$$\min_{\substack{tr(A)=0\\ tr(A^2)=c_2-c_1^2/3}}\quad AK_4A+\frac{2}{3}c_1AK_4\cdot I+\frac{1}{9}c_1^2I\cdot K_4\cdot I\cdot \qquad (3)$$

where

$$K_4 = K_1 - K_2^*\, K_3^{-1}\, K_2.$$

To see how we can find this last minimum, let the variable v vary over the space of all matrices of zero trace, and let v_o designate the matrix $c_1\,K_4\,I/3$. Let $c_3 = c_2 - c_1^2/3$ and $c_4 = c_1{}^2\,IK_4\,I/9$. Then the minimum (3) can be written as

$$\min_{|v|^2=c_3} \; vK_4v+2v_o\cdot v+c_4 \tag{4}$$

By Lagrange's multiplier principle the minimizing vector v satisfies $K_4\,v + v_0 - \lambda\,I\,v = 0$ for some real λ, in addition to the equation $|v|^2 = c_3$. Thus we have $v = (\lambda I - K_4)^{-1}v_o$. (However, if v_o is orthogonal to the eigenvector c of K_4 belonging to a particular eigenvalue μ (in particular, if $v_o = 0$, which will always be the case if $c_1 = 0$) we could also have $v=(\mu-K_4)^{-1}\;v_o + \alpha e$ where α must be chosen to make $|v|^2 = c_3$.)

Thus (assuming that the eigenvalues of the positive definite symmetric matrix K_4 are simple) we need only examine a finite number of possibilities in searching for the v which realizes the minimum (4): namely these cases connected with eigenvalues μ of K_4 noted just above, plus the roots λ of the equation $|(\lambda I - K_4)^{-1}\,v_o\,|^2 = c_3$. This last equation is algebraic and of degree equal to the dimension of the space 3 by 3 of matrices of trace zero, i.e. is of degree 10.

Little change in the above discussion is needed to cover the case in which P (hence Q) is of even degree $2k+2$. Here we have only to replace (1'') by

$$\min_{\substack{\Delta^{k+1}Q=c_1 \\ \Delta\sum\frac{\partial}{\partial x_i}(\Delta^k Q)^2=c_2}} \quad \sum w_l\left(Q(x_l)\right)^2 \tag{1'''}$$

Introducing $Q_1 = \Delta^k\,Q$, so that Q_1 is simply a quadratic polynomial, we can write this last minimum as

$$\min_{\substack{\Delta Q_1=c_1, \\ \Delta\sum\left(\frac{\partial}{\partial x_i}Q_1\right)^2=c_2}} \quad \min_{\Delta^k Q=Q_1} \; \sum w_l\left(Q(x_l)\right)^2. \tag{5}$$

The 'inner' minimum in (5) has the abstract form

$$\min_{Tq=a} \quad qKq, \tag{6}$$

where T is a linear transformation Δ^k from the space of polynomials of order $2k+2$ to the space of polynomials of order 2 , and K is a positive definite matrix. Note that T maps its domain onto its range; to see this, we have only to note that the polynomial x^2, whose rotated versions generate the whole space of quadratic polynomials, belongs to the range of Δ^k, as is

clear since $\Delta^k x^{2k+2} = \frac{1}{2}(2k+2)!\, x^2$. Putting $K^{1/2} q = r$ in (6), we can rewrite (6) as

$$\min_{TK^{-1/2} r = a} |r|^2 = \min_{T_1 r = a} |r|^2 \tag{7}$$

where $T_1 = TK^{-1/2}$. The minimizing r in (7) is clearly the unique solution of $T_1 r = a$ which is orthogonal to the space of all vectors v satisfying $T_1 v = 0$. We can express this v as $r = T_1^* (T_1 T_1^*)^{-1} a$, since then we have $T_1 r = (T_1 T_1^*)(T_1 T_1^*)^{-1} a = a$, and moreover if $T_1 v = 0$ we have $r \cdot v = T_1^* (T_1 T_1^*)^{-1} a \cdot v = (T_1 T_1^*)^{-1} a \cdot T_1 v = 0$. It follows that the minimizing q in (6) is $K^{-1/2}\, r = K^{-1/2} K^{-1/2} T^* (TK^{-1} T^*)^{-1} a = K^{-1} T^* (TK^{-1} T^*)^{-1} a$, recovering the well known fact that the minimum (6) is just

$$a\,(TK^{-1} T^*)^{-1}\, TK^{-1} K\, K^{-1} T^*\, (TK^{-1} T^*)^{-1} a = a\,(TK^{-1} T^*)^{-1} a.$$

This shows that the minimum (5) can be expressed in terms of the polynomial Q_1 by minimum of the form that we have already treated, namely

$$\min_{\substack{\Delta Q_1 = c_1 \\ \Delta \sum \left(\frac{\partial}{\partial x_i} Q_1\right)^2 = c_2}} Q_1 K_1 Q_1, \tag{8}$$

where the symmetric matrix K is obtained in an elementary way from the matrix of the quadratic form appearing in (5), namely by two matrix inversions and two matrix multiplications. Finally, the minimum (8) can be found in the manner already explained.

When we apply the technique just described to determine the position of an object O several of whose surfaces may be visible, it is essential that we avoid mixing observed points drawn from one of O's surfaces with points drawn from some other surface of O. If O's various surfaces are separated cleanly enough by edges at which the tangent surface to O changes sharply, it may be possible to do this by applying an edge operator to an intensity image superimposed on a depth image, or directly to a depth image to bring out 'jump' boundaries. (Only components not broken by edges of either of those types should be used for matching. Note also that the matching method that we have described is likely to give immediate warning of attempts to match such 'mixed observations', since such attempts are not likely to generate small minima (5), given that the two parameters c_1, c_2 appearing in (5)

are both taken directly from one of the finite collection of model surfaces which would match the scene perfectly). However, cases in which there are smooth transitions between surfaces of O described by spline surfaces, satisfying very different equations but constructed to achieve smooth joins, will of course not be separable by any visible edges. In such cases the following approach may be useful. Let one surface satisfy a polynomial equation $P_1(x) = 0$, while the second surface satisfies $P_2(x) = 0$. Then the union of the two surfaces satisfies $P_1 P_2(x) = 0$, and can therefore be matched to the higher-degree polynomial $P_1 P_2$. A similar observation applies to corners at which three or four surface patches come together.

Sets of observed points for use in the matching procedure that has been described can be gathered in any convenient way. For example, one can gather partial depth information by illuminating a scene with just one plane of light, which gives the depth of all object points lying in the illuminated plane. The resulting curve can broken at all its sharp (particularly concave) corners, and again at all sharp jumps in intensity, yielding curve sections each of which probably lies on one single object surface. This same procedure can then be repeated for a second plane orthogonal to the original plane of illumination and cutting it at a point well interior to one of the aforementioned curve sections, or, still better, repeated for planes forming a pair of orthogonal grids. Any two 3-D curve sections likely to lie together in one single object surface then 'predict' the orientation of such a surface.

Once the presence and orientation of a particular object surface has been predicted, the position in space of the body containing this surface can be determined, and then the prediction can be checked by comparing an artificially generated depth image of the body to the observed scene and looking for substantial regions of agreement. Such regions should of course represent other surfaces of the same body. (It can also be verified that no point in the scene actually observed lies substantially behind any surface of a body conjectured to be present). Once all regions of an observed scene have been identified as object surfaces in this manner, all objects present and visible in the scene will be known.

At this point, the further task of following the motion of these objects becomes substantially easier, and can be handled as follows: for each visible body, one can take points well within the visible surface portions $P(x) = 0$. Suppose that the initial position of each such portion $P(x) = 0$ is known, and suppose that after a small interval of time each of the points of this surface portion has moved from an original position x to a new position $Rx+b$, where $R=I+a$ is a small rotation and b is a small translation. Suppose that points x_l on the body surface in its new position are observed. Then the points $R^{-1}x_l - b$ lie on the original body surface. For each such point, take any sufficiently close point y_l on the original body surface (as known from its equation as first estimated), for example let y_l be the point on the original surface whose projections along two coordinate axes agree with those of x_l; these axes must be chosen so that the third coordinate axis intersects the surface $P(x)=0$ at a substantial angle near x_l. Then the difference vector $R^{-1}x_l - b - y_l = x_l - y_l + ax_l - b$ lies close to the plane tangent to $P(x)=0$ at y_l , and hence can be taken to satisfy the equation

$$L_l\,(x_l - y_l + ax_l - b) = 0,$$

where $L_l = \partial P(y_l)$ is the vector normal to $P(x) = 0$ at y_l. These relationships give us a family of equations of which any six should suffice to determine the six linearly independent quantities $[a,b]$; of course, it is better to use more equations and then to estimate a and b using a least squares procedure. Note that much this same procedure can be used to improve the initially conjectured match of an observed body to a model body once the observed surfaces which should lie in each particular rotated/translated model body have been identified. Note also that if the manner in which a body's shape can change is simple and known (as in the case of an expanding sphere, or of an ellipsoid gradually changing in eccentricity), a similar simple technique can be used to track shape changes as well as rotations.

It is also worth observing that the procedure that we have described applies individually to each of the observed surfaces in a scene containing multiple bodies, provided only that the general form of these surfaces is known (e.g. it may be known that every such surface is

either plane or quadric), even if the overall way in which these surfaces cohere to form bodies is not known initially. Suppose, for example, that the observed scene is known to contain only one body. Then by matching the surfaces visible from one viewpoint, we can define part of the body shape, and by doing this from sufficiently many viewpoints can hope to acquire a complete understanding of the object. The resulting object description would consist of a list of equations $P_i(x) = 0$ (or more precisely of inequalities $P_i(x) \geq 0$) defining all the surfaces of the body (the side of these surfaces on which the interior of the body lies being described by the aforementioned inequalities), together with a set of lists, each of which defines the body surfaces which come together at one of the vertices v of the body, in the circular order of these surfaces around the vertex v.

Development and demonstration of a capability to acquire the shapes of manufactured objects directly from combined depth and intensity image observations would represent a significant step forward in computer vision technology.

References

[ABBF84] N. Ayache, J. Boissonat, B. Bollack, and B. Faverson, Automatic Handling of Overlapping Workpieces, paper presented at Int. Conf. Pattern Recognition, Montreal, 1984.

[BS84] M. Bastuscheck and J.T. Schwartz, Preliminary implementation of a ratio depth sensor, Tech. Rept. 124, Computer Science Dept., Courant Institute, N.Y.U., June 1984.

[BC82] R. Bolles and R. Gain, Recognizing and Locating Partially Visible Objects: the Local Feature-Focus Method. Int. J. Robotics Res. v.1, (1982). pp.57-82.

[BHH84] R. Bolles, P. Horaud, and M. Hannah, 3DPO: A Three Dimensional Part Orientation System. In: Robotics Research, the First International Symposium. M. Brady and R. Paul, eds., MIT Press, 1984, pp. 413-424.

[Ch82] B. Chazelle, A theorem on polygon cutting with applications, Proc. 23rd IEEE Symp. on the Foundations of Computer Science, 1982, pp. 339-349.

[GLP85] W. Grimson and T. Lozano-Perez, Recognition and Localization of Overlapping Parts from Sparse Data in Two and Three Dimensions, 1985 IEEE Int. Conf. on Robotics and Automation, St. Louis, pp. 61-66.

[GLP84] W. Grimson and T. Lozano-Perez, Model-Based Recognition and Localization from Sparse Range or Tactile Data, Int. J. Robotics Res. v.3 (1984), pp. 3-35.

[KSSS85] A. Kalvin, E. Schonberg, J.T. Schwartz, M. Sharir, Two Dimensional Model Based Boundary Matching Using Footprints, Tech. Rept., Computer Science Dept., Courant Institute, N.Y.U., 1985.

[KK85] M. Koch and R. Kashyap, A vision System to Identify Occluded Industrial Parts, 1985, IEEE Int. Conf. on Robotics and Automation, St. Louis, pp. 55-60.

[LP84] D.T. Lee and F.P. Preparata, Euclidean shortest paths in the presence of rectilinear barriers, *Networks* 14 (1984) pp. 393-410.

[Pa84] C.H. Papadimitriou, An algorithm for shortest path motion in three dimensions, preprint, Computer Science Dept., Stanford University, July 1984.

[S83] J.T. Schwartz, Structured light sensors for 3-D robot vision, Tech. Rept. 65, Computer Science Dept., Courant Institute, N.Y.U., March 1983.

[SS84a] J.T. Schwartz and M. Sharir, Some remarks on robot vision, Tech. Rept. 119, Computer Science Dept., Courant Institute, N.Y.U., April 1984.

[SS84b] M. Sharir and A. Schorr, On shortest paths in polyhedral spaces, *Proc. 16th ACM Symp. on Theory of Computing*, 1984, pp. 144-153.

[TMV85] J. Tunney, T. Mudge, and R. Volz, Recognizing Partially Hidden Objects, 1985, IEEE Int. Conf on Robotics and Automation St. Louis, pp. 48-54

[BB82] D. Ballard and C. Brown, Computer Vision, Prentice-Hall Publishers, Englewood Cliffs, N.J., 1982.

8.24.89	~~ILL - NJIT~~

~~ILL 10.23.87~~			
ILL- 8.24.89			